Benjamin O'Neill

Efficient Asymmetric Cryptography for RFID Access Control

Computer Science Tripos - Part II

Emmanuel College

2018

Proforma

Name:	Benjamin O'Neill
College:	Emmanuel College
Project Title:	Efficient Asymmetric Cryptography for RFID Access Control
Examination:	Computer Science Tripos Part II - 2018
Word count:	TODO
Project Originator:	Dr Markus Kuhn
Project Supervisor:	Dr Markus Kuhn

Original aims of the project

TODO: Summarise project proposal.

Work completed

TODO:

Special difficulties faced

TODO: Maybe mention library incompatibilities, broken cards.

Declaration of Originality

I, Benjamin O'Neill of Emmanuel College, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.

Signed, Benjamin O'Neill

Date: TODO

Table of Contents

TODO
Look into how to link to things using LaTeX.

1 Introduction

TODO
Motivation for the project. What needs are being addressed.
Crypto description, symmetric and symmetric.
Background information about smartcards.
current university smartcard system, shortcomings. (Look into MIFARE protocol).
Detail security flaws in MIFARE classic. Uni may be moving to MIFARE plus.
Information about JavaCard and development process.

2 Preparation

TODO
Work undertaken before code was written. How proposal was refined and clarified.
Analysis of problems with MIFARE classic. Outline main problems.
Quick analysis of different protocols and comparisons.
Reading about smartcard and java card stuff. ISO7816 standards to get background understanding, descriptions of java card on oracle website.
APDU stuff
Research into different smartcards, readers etc.
Researched asymmetric cryptography, and when OPACITY chosen, researched elliptic curve cryptography.
"Requirements analysis" section where project proposal points elaborated. Reference other software engineering techniques.
Familiarised with various misc things like ASN1, had to learn various quirks and limitations of restricted Java Card language (e.g. int-to-short casting, restrictions on APDU passing, different garbage collection system). Very little online help about any of this.
Analysis of available tools, with reference to different JC versions (eclipse plugin for 3.0.5 most common, ant/command line more common in older ones. Used both ANT and command line when appropriate with 3.0.4)
Also GPShell, specific smartcard details, versions purchased.

3 Implementation

TODO
What was actually produced. Programs/code written.
Could move a few things to here depending on whether learning the dev process counts as preparation or implementation.
Give examples of useful code abstractions.
Details of design decisions, libraries used, code produced (i.e. non-JCMathLib Java Card code including CMAC and ECDSA. (TODO: implement ECDSA)
Use small code fragments to illustrate certain things, but not too much here.
Properly point out JCMathLib and other 3rd party things. Point out that license is compatible with the context I am using it in.
Can highlight major milestones in the process.
How is card issued? Prob issuer gets Id, adds it to allowed IDs of all terminals. TODO: Is this right?
XML mock PB store.

4 Evaluation

TODO
Assessors look for signs of success, thorough and systematic evaluation. (see pink book 8.3)
Give sample output, timing, perhaps photographs of current system with discussion of how it would be implemented in reality.
APDU trace
Runtime for initial encounter vs later counter using PB
Analysis of how much time each operation actually takes.
Were original goals achieved?
Do a brief security analysis of the system, considering security of the protocol itself, and other things e.g. whether issuing process has faults, whether PB entries can be altered maliciously etc.
Some residual bugs expected. If any exist, explain briefly. Show it still works in the basic case.
What general shortcomings does it have?

5 Conclusions

TODO
Don't introduce new information here, just summarise and briefly discuss.
What I have learned generally about the area, and what should I do differently in future when doing projects.
Should be short, refer to introduction. How would project have been planned or executed differently if done again.
What more could be done.

6 Bibliography

TODO

7 Appendices

TODO
Sample code, protocol diagram and other details, other details about the workings of smartcards e.g. from ISO7816, Java Card documentation.

8 Index

TODO (Optional)